

CLAUDE SLASH COMMANDS

Every built-in command, what it does, and how to use it.

✱ CLAUDE

COMMAND	WHAT IT DOES	HOW TO USE IT
<code>/model</code>	Switch the active model for this session	<code>/model</code> to open the picker, or <code>/model <name></code> to jump straight to one (e.g. opus)
<code>/compact</code>	Summarize the conversation to free up context	Run <code>/compact</code> any time, or <code>/compact <instructions></code> to steer what's kept
<code>/goal</code>	Set a completion condition to work toward across turns	<code>/goal <condition></code> to set it, <code>/goal clear</code> to remove it – pairs well with <code>/plan</code>
<code>/plan</code>	Have Claude propose a plan before making changes	<code>/plan <task></code> before a risky edit or refactor, then approve or adjust it
<code>/mcp</code>	Manage Model Context Protocol server connections	<code>/mcp reconnect <server></code> , <code>/mcp enable <server all></code> , or <code>/mcp disable <server all></code>
<code>/clear</code>	Wipe the conversation and start over with empty context	Run <code>/clear</code> for a blank slate – history isn't recoverable after
<code>/cost</code>	Show token usage and an estimated dollar cost	Run <code>/cost</code> any time – the estimate is local, not your invoice
<code>/permissions</code>	Review or edit the tool allow / ask / deny rules	Run <code>/permissions</code> to open the rules editor
<code>/agents</code>	Points you to your subagent setup	Run <code>/agents</code> for a reminder – subagents are configured as files in <code>.claude/agents/</code>
<code>/memory</code>	Edit CLAUDE.md and what Claude remembers	Run <code>/memory</code> to open the project or user memory files
<code>/init</code>	Generate a starting CLAUDE.md for a new project	Run <code>/init</code> in a fresh repo – Claude scans the structure and drafts it

🔗 QUICK START

- 1 Type `/` to open the command list – keep typing to filter.
- 2 Add an argument after a space, e.g. `/model opus`.
- 3 Run `/help` to see every command in your version.
- 4 Add custom commands as files in `.claude/commands/`.

💡 TIPS

- `/goal` pairs well with `/plan`: shape the objective first, then lock it in.
- Run `/compact` before hitting the context limit – you control what's kept.
- `/clear` is destructive and unrecoverable. Save what you need first.
- `/permissions` rules are per-project and apply to anyone who opens it.

CHATGPT SLASH COMMANDS

Every built-in command, what it does, and how to use it.

CHATGPT

COMMAND	WHAT IT DOES	HOW TO USE IT
<code>/model</code>	Pick the model for this chat	Run <code>/model</code> , then choose from the list (e.g. <code>gpt-5.4</code> , <code>gpt-5.5</code>)
<code>/compact</code>	Shrink context once a session is running long	Run <code>/compact</code> mid-chat once things feel slow or repetitive
<code>/goal</code>	Set a persistent goal to work toward	Run <code>/goal</code> , then describe it – pair it with <code>/plan</code> first to shape the goal
<code>/plan</code>	Toggle multi-step planning mode on or off	Run <code>/plan</code> before a multi-step task to have it plan first
<code>/mcp</code>	Check connected MCP server status and tools	Run <code>/mcp</code> to see what's connected, or <code>/mcp verbose</code> for full diagnostics
<code>/project</code>	Set the project new chats will use	Run <code>/project</code> , then pick one – it applies to chats you start next, not the current one
<code>/status</code>	See chat ID, context usage, and rate limits	Run <code>/status</code> any time to check where the session stands
<code>/memories</code>	Configure whether it uses or generates memories	Run <code>/memories</code> to turn memory on/off or review what's stored
<code>/personality</code>	Choose the response style	Run <code>/personality</code> , then pick friendly, pragmatic, or none
<code>/apps</code>	Browse connected apps to bring into a prompt	Run <code>/apps</code> , pick one, and it's inserted as <code>\$app-slug</code>

QUICK START

- 1 Type `/` to open the command list, then keep typing to filter.
- 2 Start with `/plan` to refine a goal, then set it with `/goal`.
- 3 In Codex CLI, run `/mcp` to inspect available tools.
- 4 Saved custom prompts appear as `/prompts:<name>`.

TIPS

- `/goal` starts Goal mode — use the progress row to pause, resume, edit, or clear it.
- `/plan` comes before `/goal`: shape and refine the objective, then lock it in.
- `/project` sets what new chats will use going forward — not the current one.
- `/compact` shrinks context once a session runs long or feels slow.